

Apellidos:

Nombre:

Matrícula:

---

# Examen

## 105000016 - Programación para Sistemas Grado en Ingeniería Informática

Lenguajes y Sistemas Informáticos e Ingeniería de Software

Facultad de Informática

Universidad Politécnica de Madrid

Curso 2013/2014 - Junio 2014

### Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **una hora**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.
- La fecha prevista de publicación de calificaciones es el **23 de junio**, y se realizará a través del Aula Virtual de la asignatura.
- La revisión del examen tendrá lugar el **25 de junio** a las 12:00 en la sala (se informará).

### Cuestionario

- (1 punto) 1. Escribir un mandato *bash* que muestre por la salida estándar todos los ficheros del directorio de trabajo junto con, al menos, sus correspondientes permisos (puede mostrar más información sobre cada fichero).

**Solución:** `ls -al`

- (1 punto) 2. Dar un mandato *bash* que guarde en la variable **hoy** la fecha del momento en que se ejecute (no importan ni el formato ni si también guarda la hora)

**Solución:** `hoy=$(date)`

- (1 punto) 3. Estando el usuario en un directorio en el que tiene permiso de escritura, dar un mandato (o mandatos) *bash* para que copie el fichero **viejo** de ese mismo directorio en el fichero **nuevo** en ese mismo directorio solamente en el caso en que sea posible hacerlo (y si no es posible ni lo intente). No existe ningún fichero llamado **nuevo** en dicho directorio.

**Solución:** `[ -r viejo ] && cp viejo nuevo`

- (1 punto) 4. En el manual de Bash, en la sección "Expansión de Parámetros", se puede leer:

```
${variable:+palabra}
```

Si *variable* es null o no está definida entonces se sustituye por nada, en otro caso se sustituye por *palabra*.

¿Cuál será la salida estándar tras la ejecución de los siguientes mandatos?

```
unset A
unset B
unset C
B=
C=Hola
echo A: ${A:+Adios}
echo B: ${B:+Adios}
echo C: ${C:+Adios}
```

**Solución:**

```
A:
B:
C: Adios
```

- (1 punto) 5. Dado el siguiente código C , que mostraría por la salida estándar

```
#include<stdio.h>
int main()
{
    int i=0, suma=0;

    while (i<=5) {
        suma +=i;
        i++;
    }
    printf("SUMA_EN_LA_ITERACIÓN_ %d_VALE_ %d\n", i, suma);
    return 0;
}
```

**Solución:** SUMA EN LA ITERACIÓN 6 VALE 15

- (1 punto) 6. Escribe la salida que produciría el siguiente fragmento de código

```
int contador =10 ;
int *temporal, suma=0;
temporal=&contador;
*temporal=20;
temporal=&suma;
*temporal=contador;
printf ("_ %d_ %d_ %d\n", contador, *temporal, suma);
```

**Solución:**

20 20 20

- (1 punto) 7. Escribir como compilarías un programa llamado ejemplo.exe que está compuesto por dos códigos fuentes: principal.c y ayuda.c. Teniendo en cuenta que ambos utilizan funciones de la librería matemática de C.

**Solución:** gcc -o ejemplo.exe principal.c ayuda.c -lm

- (1 punto) 8. Indica el comando gdb que nos permitiría parar la ejecución del programa que estamos depurando cuando se modifique el valor de una variable

A. break      B. **watch**      C. display      D. print

- (1 punto) 9. Escribe la salida que produciría el siguiente fragmento de código

```
int *p,x,*p2;
int a[5] = {1,2,3,4,5};
x=5;
p=&x;
p2=a;
*(p2+1)=*p;
printf ("%d_ %d_ %d_ %d_\n",x,*p,*p2,a[1]);
```

**Solución:** 5 5 1 5

- (1 punto) 10. Escribe las funciones que se llamarían tras la ejecución del siguiente fragmento cuando c toma un valor de 3

```
switch (c) {
    case 1:
    case 2: Funcion2 ();
    case 3: Funcion3 ();
    case 4: Funcion4_1 ();Funcion4_2 ();break;
    case 5: Funcion_5 ();
    default: FuncionX ();
}
```

**Solución:** Funcion3, Función4.1 y Función4.2

- (1 punto) 11. ¿Cuál de las siguiente funciones C no requiere la petición de memoria dinámica?  
A. realloc    B. calloc    **C. strcpy**    D. strdup
- (1 punto) 12. Declarar un doble puntero para manejar una matriz dinámica que va almacenar números reales. A continuación hacer una reserva de memoria dinámica para una matriz de 3 filas y 5 columnas.

**Solución:**

```
#include<stdio.h>
int main()
{double **matriz;
 int i;
 matriz=(double **) malloc(sizeof(double*)*3);
 for (i=0;i<3;i++)
     matriz[i]=(double *) malloc(sizeof(double)*5);
 return 0;
}
```